



FPT POLYTECHNIC



www.poly.edu.vn

LẬP TRÌNH ANDROID 1

TỔNG QUAN VỀ ANDROID

❑ Giới thiệu về Android

- ❖ Tổng quan về Android
- ❖ Kiến trúc
- ❖ Môi trường phát triển

❑ Giới thiệu về Android Studio

- ❖ Tạo project trên Android Studio
- ❖ Cấu trúc project
- ❖ Biên dịch và gỡ lỗi ứng dụng



FPT POLYTECHNIC



www.poly.edu.vn

LẬP TRÌNH ANDROID 1

BÀI 1.1: TỔNG QUAN VỀ ANDROID

MỤC TIÊU

- NẮM ĐƯỢC VỀ TỔNG QUAN ANDROID
- KIẾN TRÚC CỦA ANDROID
- MÔI TRƯỜNG PHÁT TRIỂN
- CẤU TRÚC PROJECT ANDROID
- BIÊN DỊCH VÀ GỠ LỖI ỨNG DUNG





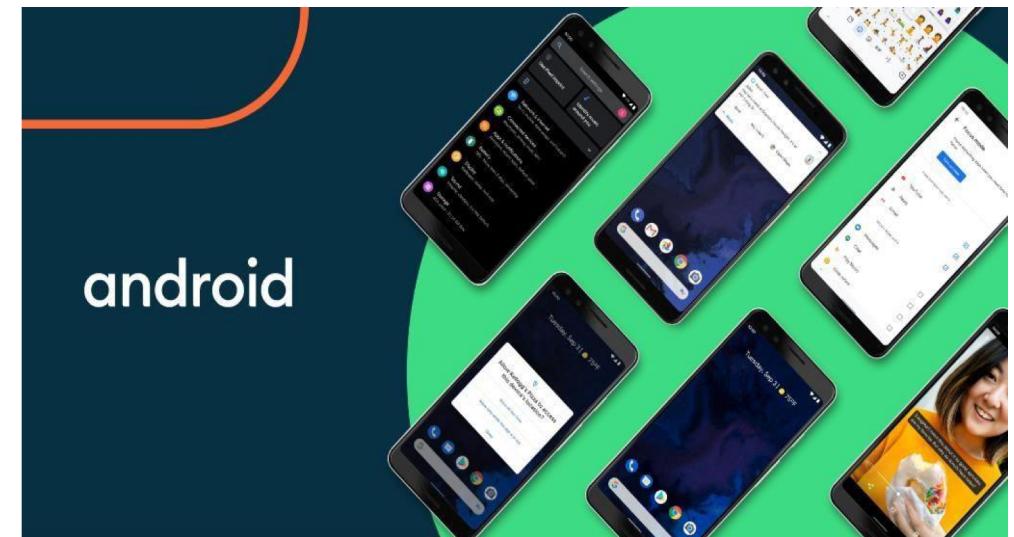
TỔNG QUAN VỀ ANDROID

❑ Hệ điều hành Android

- ❖ Là hệ điều hành nguồn mở dựa trên Linux được phát triển bởi Tổng công ty Android. Google mua lại (2005) và cho ra mắt vào năm 2007.
- ❖ Phát triển cho các thiết bị có màn hình cảm ứng chủ yếu dành cho điện thoại, máy tính bảng...

Ngôn ngữ Java (2011: Kotlin)

Sử dụng Bộ phát triển phần mềm Android (SDK).



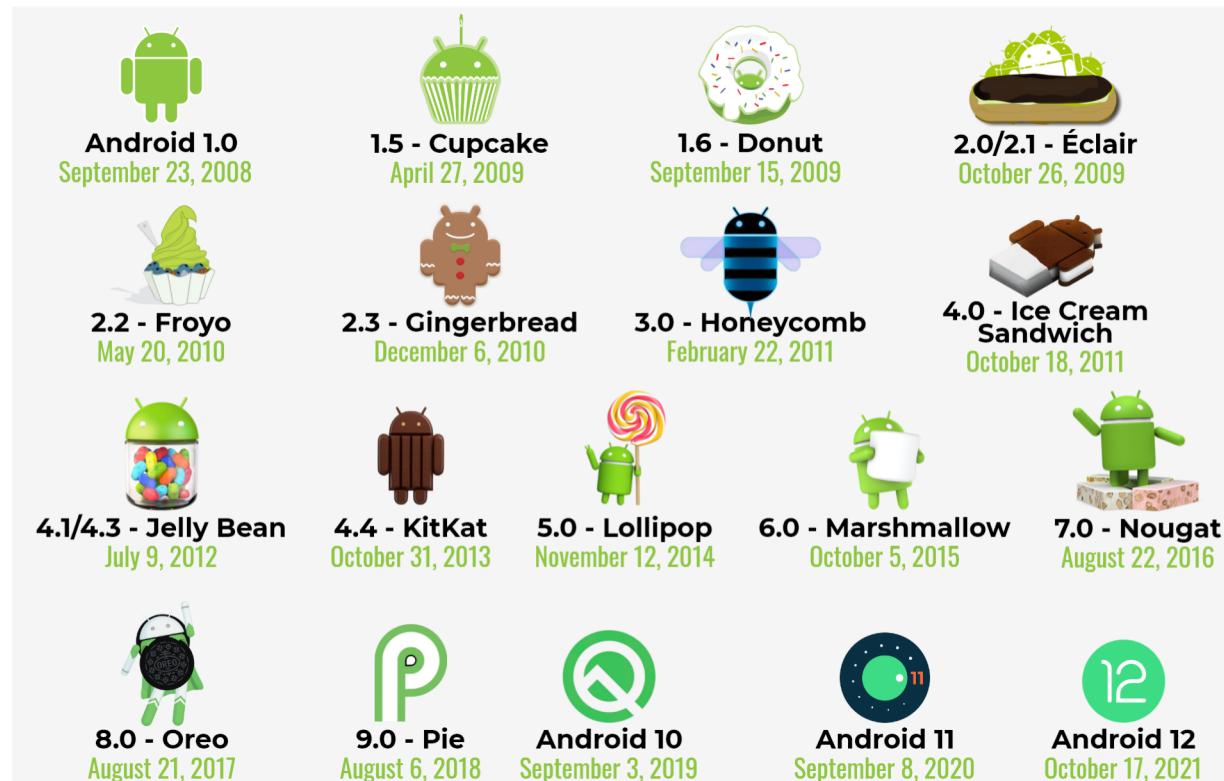
☐ Liên minh OHA (Open handset alliance)

- ❖ Liên minh Khoảng 80 công ty cùng xây dựng nền tảng mobile mới



☐ Lịch sử phát triển của Android

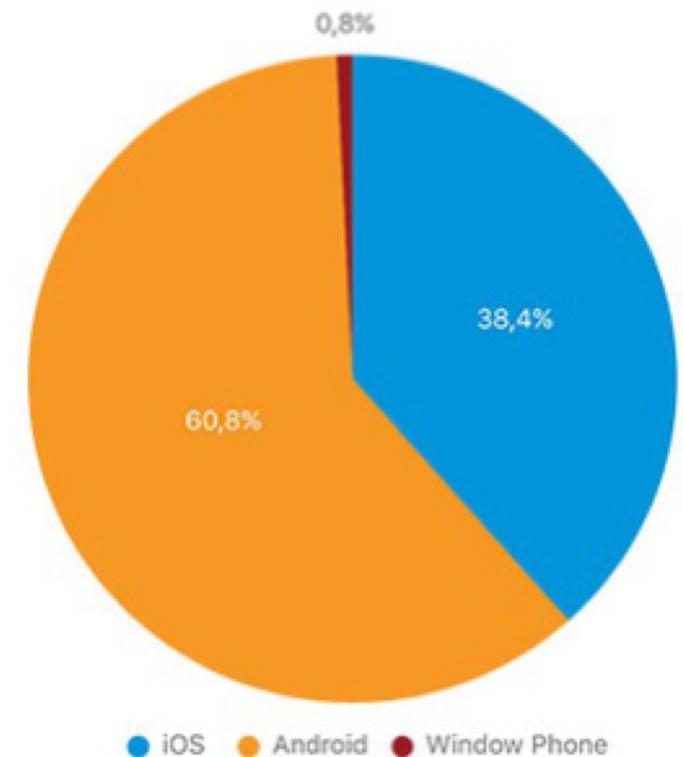
- ❖ Phiên bản hiện tại: Android 14 (Android 14 Developer Preview)
- Android 13 (August 15, 2022)
- ❖ Các phiên bản trước đó: 2008- 2021



Thị phần các hệ điều hành trên smartphone



Thị phần theo hệ điều hành Smartphone tháng 6/2020



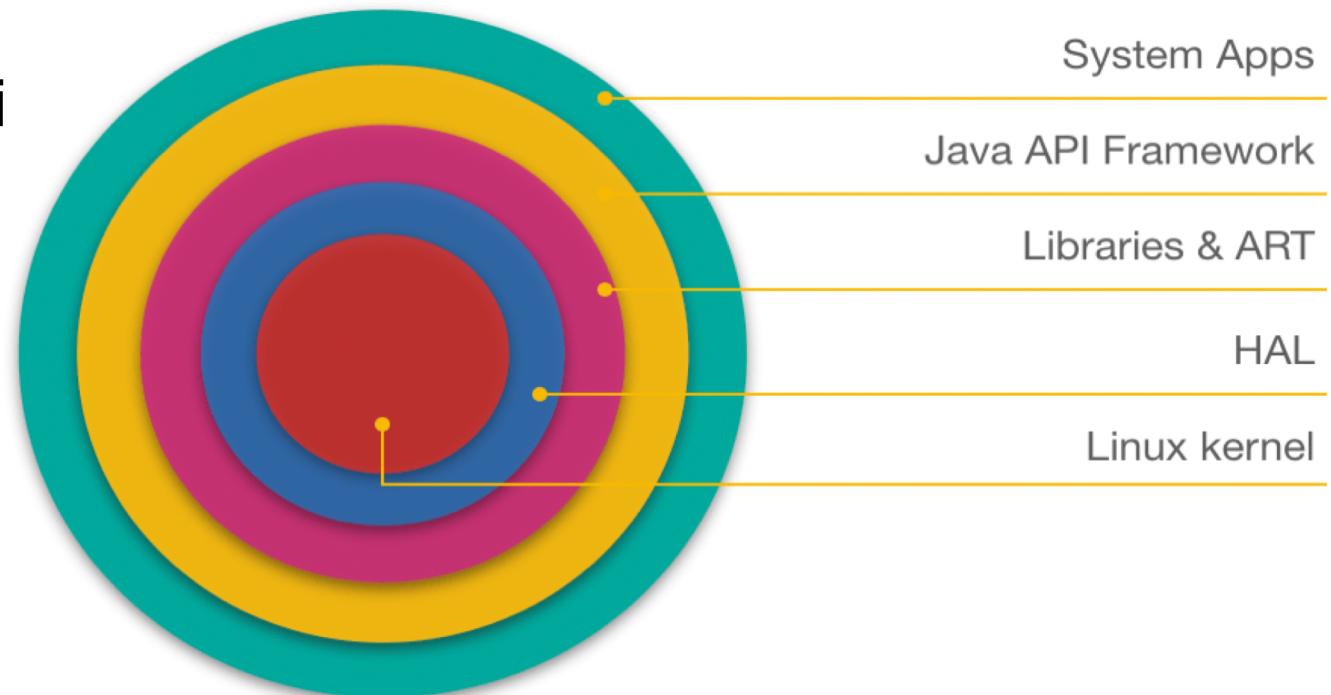


KIẾN TRÚC ANDROID

❑ **Android Platform**

Là môi trường phần mềm được xây dựng cho thiết bị di động Android bao gồm:

- ❖ Hệ điều hành dựa trên nhân Linux
- ❖ Giao diện người dùng
- ❖ Ứng dụng người dùng cuối
- ❖ Thư viện code
- ❖ Application Framework

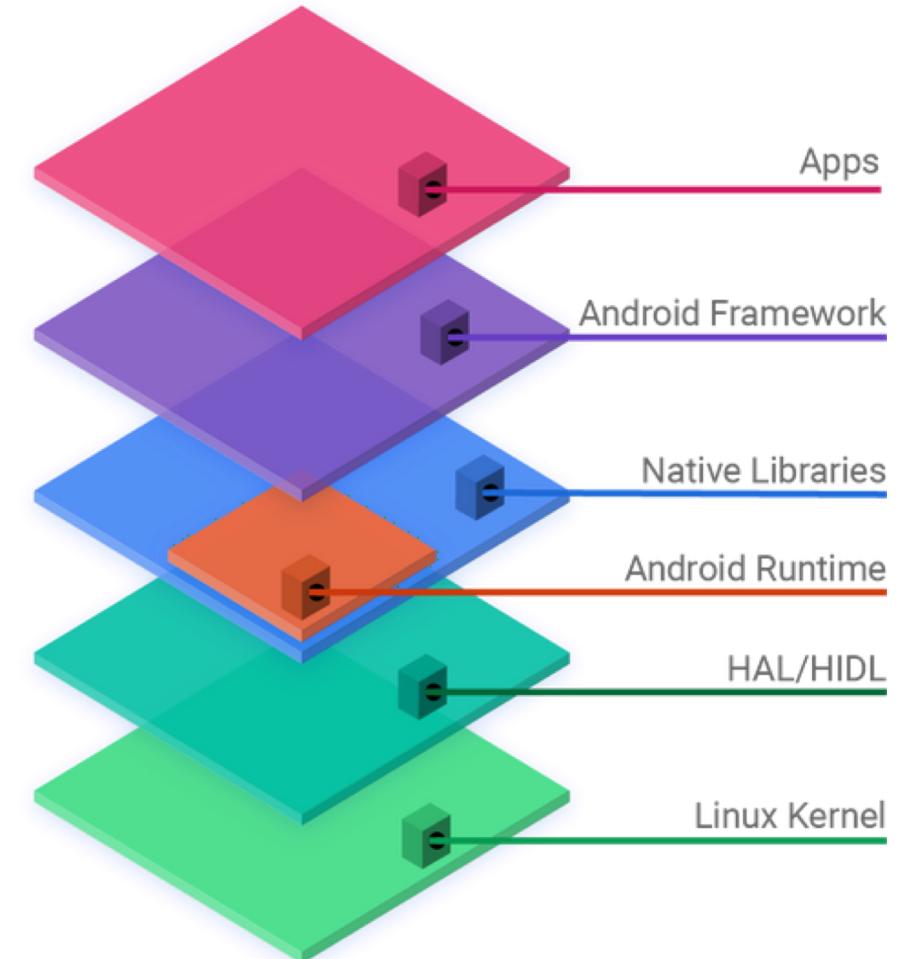


❑ Kiến trúc Android

Android là một hệ điều hành mã nguồn mở dựa trên nền tảng Linux.

Gồm có 6 tầng là:

- ❖ Linux Kernel
- ❖ Hardware Abstraction Layer – HAL
- ❖ Android Runtime
- ❖ Native C/C++ Libraries
- ❖ Java API Framework
- ❖ System Apps

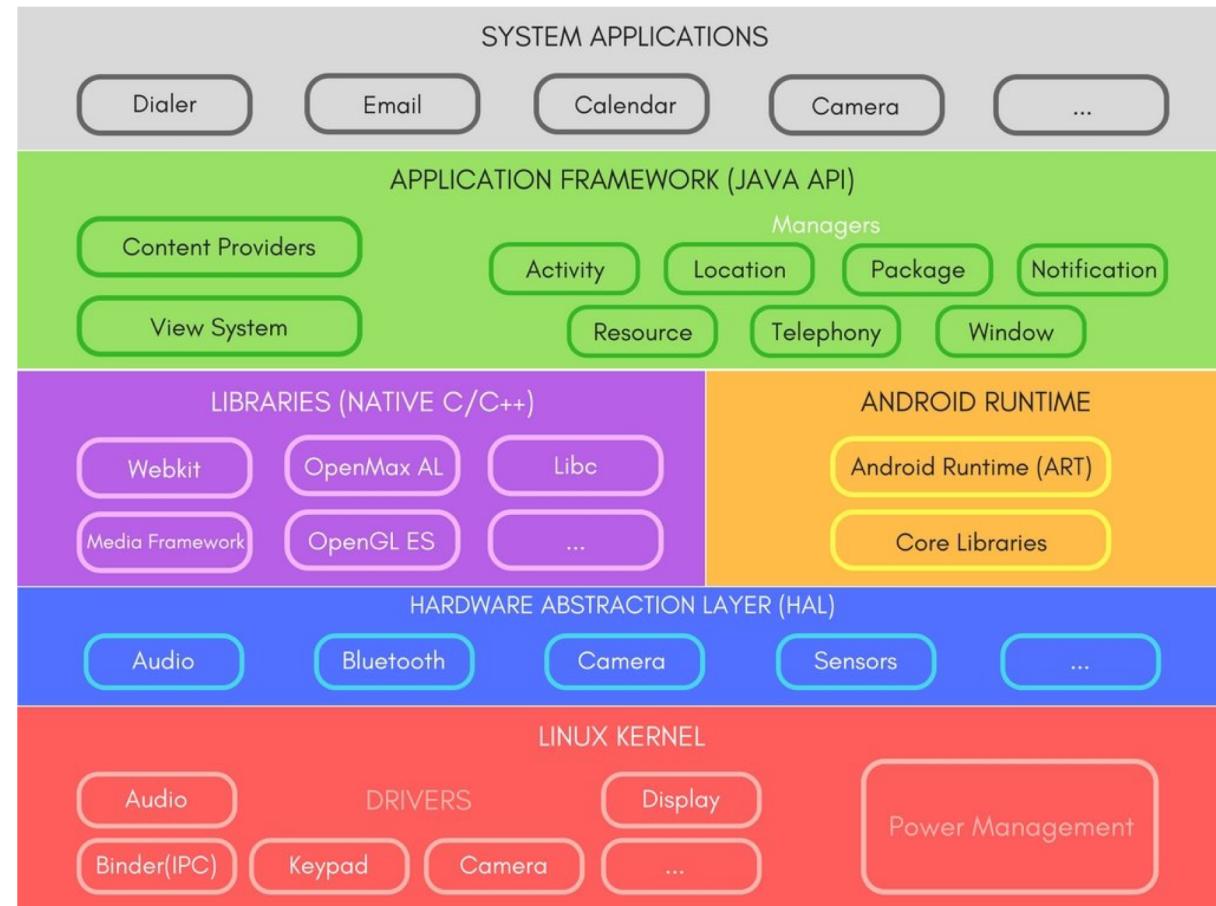


❑ Kiến trúc Android

Android là một hệ điều hành mã nguồn mở dựa trên nền tảng Linux.

Gồm có 6 tầng là:

- ❖ Linux Kernel
- ❖ Hardware Abstraction Layer – HAL
- ❖ Android Runtime
- ❖ Native C/C++ Libraries
- ❖ Java API Framework
- ❖ System Apps



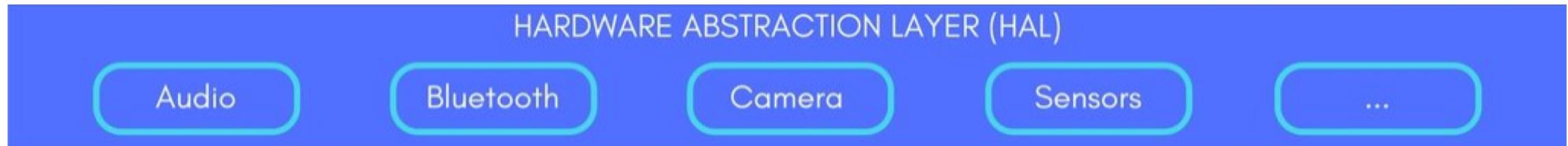
❑ Linux Kernel

Nhân Linux (Linux Kernel) là tầng dưới cùng và là nền tảng trong kiến trúc của hệ điều hành Android. Đây là Linux 3.6 với khoảng 115 bản vá. Tầng này cung cấp mức độ trừu tượng giữa phần cứng của thiết bị, và nó chứa tất cả các trình điều khiển phần cứng thiết yếu như máy ảnh, bàn phím, màn hình,... Ngoài ra, hạt nhân (kernel) xử lý tất cả những thứ mà Linux có thể làm tốt nhất như kết nối mạng và một loạt các trình điều khiển thiết bị, giúp làm giảm bớt khó khăn khi giao tiếp với phần cứng ngoại vi.



❑ Hardware Abstraction Layer – HAL

- ❖ Lớp trừu tượng phần cứng (Hardware Abstraction Layer – HAL) là tầng thứ hai tính từ dưới lên trong kiến trúc của hệ điều hành Android. Tầng này sử dụng khung API (Application Programming Interface) Java cấp cao hơn để cung cấp các giao diện tiêu chuẩn, thực hiện các khả năng của phần cứng thiết bị.
- ❖ HAL bao gồm nhiều mô-đun, mỗi mô-đun sẽ thực hiện một giao diện cho một loại thành phần phần cứng cụ thể, ví dụ như mô-đun camera hoặc mô-đun bluetooth.
- ❖ Khi API thực hiện việc truy cập vào phần cứng của thiết bị, hệ điều hành Android sẽ tải mô-đun cho thành phần của phần cứng đó.



❑ Android Runtime

- ❖ Với những thiết bị chạy Android phiên bản 5.0 (API cấp 21) trở lên, mỗi ứng dụng sẽ có quy trình chạy riêng và với thời gian chạy Android (Android Runtime – ART) riêng.
- ❖ Tầng này cung cấp một thành phần chính được gọi là máy ảo Dalvik (Dalvik Virtual Machine – Dalvik VM), đây là một loại máy ảo Java được thiết kế và tối ưu hóa đặc biệt cho Android.
- ❖ Dalvik VM sử dụng các tính năng cốt lõi của Linux như quản lý bộ nhớ và đa luồng, cho phép mọi ứng dụng Android chạy trong quy trình riêng với phiên bản Dalvik VM của riêng nó.
- ❖ ART cũng cung cấp một bộ thư viện cốt lõi, cho phép các nhà lập trình viết các ứng dụng Android bằng ngôn ngữ lập trình Java tiêu chuẩn.

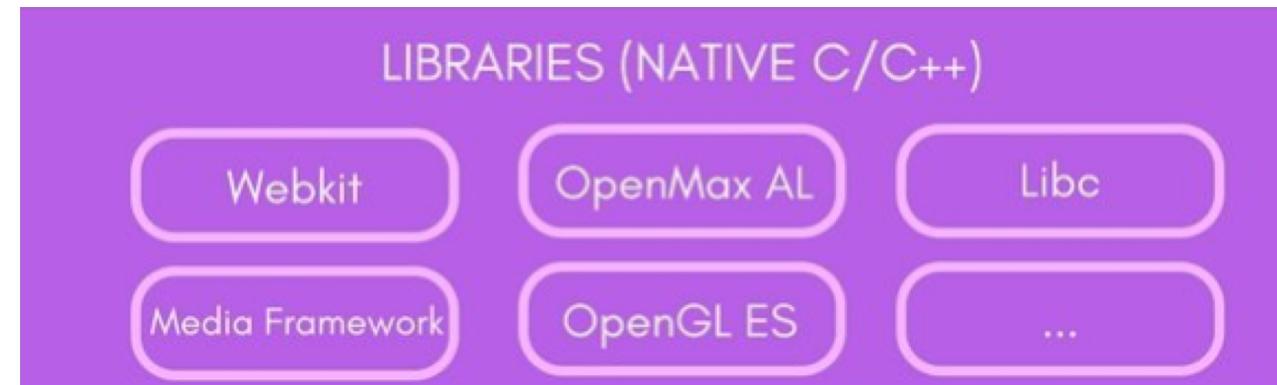


❑ Native C/C++ Libraries

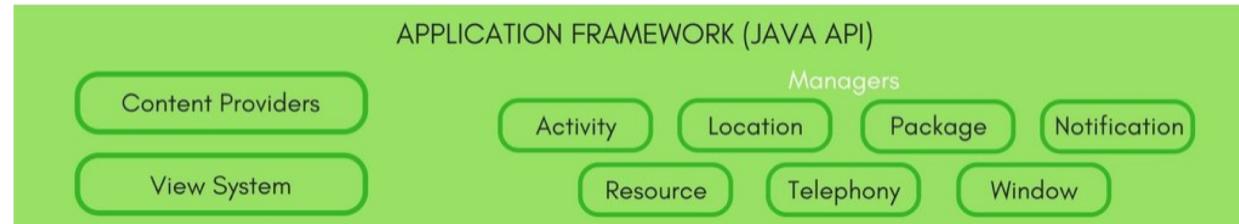
- ❖ Những thành phần và dịch vụ hệ thống cốt lõi như ART và HAL được xây dựng từ mã gốc, được viết bằng ngôn ngữ C và C++. Android cung cấp các khung API Java để hiển thị chức năng của một số thư viện C/C++ gốc (Native C/C++ Libraries) cho các ứng dụng.

☞ **Ví dụ:** Chúng ta có thể thêm thao tác hỗ trợ vẽ và đồ họa 2D và 3D cho ứng dụng bằng cách truy cập OpenGL ES (Open Graphics Library for Embedded System) thông qua API Java OpenGL.

- ❖ Nếu bạn đang phát triển một ứng dụng yêu cầu mã C/C++, có thể sử dụng Android NDK (Native Development Kit) để truy cập trực tiếp vào một số thư viện C/C++ gốc từ mã gốc của mình.



☐ Java API Framework



- ❖ Java API Framework cung cấp nhiều dịch vụ cấp cao hơn dưới dạng các lớp Java dành cho các ứng dụng. Các nhà lập trình được phép sử dụng các dịch vụ này trong các ứng dụng của họ.
- ❖ Các dịch vụ chính của Java API Framework bao gồm:
 - ❑ Trình quản lý hoạt động (Activity Manager): Kiểm soát tất cả các khía cạnh trong vòng đời của ứng dụng và activity stack.
 - ❑ Nhà cung cấp nội dung (Content Providers): Cho phép ứng dụng xuất bản và chia sẻ dữ liệu với các ứng dụng khác.
 - ❑ Trình quản lý tài nguyên (Resource Manager): Cung cấp quyền truy cập vào các tài nguyên không code như chuỗi, cài đặt màu, bố cục giao diện người dùng.
 - ❑ Trình quản lý thông báo (Notifications Manager): Cho phép ứng dụng hiển thị những cảnh báo và thông báo đến người dùng.
 - ❑ Hệ thống xem (View System): Tập hợp nhiều dạng khung hình có thể mở rộng, sử dụng để tạo giao diện cho người dùng ứng dụng.

☐ System Apps

Tầng trên cùng trong kiến trúc của hệ điều hành Android là System Apps. Chúng ta sẽ tìm thấy tất cả các ứng dụng mà nhà sản xuất tạo ra sẽ được cài đặt trên tầng này. Ví dụ như Contacts Books, Browser, Games...

Trên đây là bài viết chia sẻ của FUNiX về Kiến trúc các tầng của hệ điều hành Android. Hy vọng những kiến thức này sẽ có ích cho các bạn trong quá trình học tập và theo đuổi nghề nghiệp. Cảm ơn các bạn đã đón đọc.



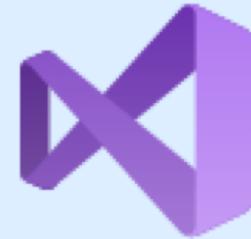


MÔI TRƯỜNG PHÁT TRIỂN

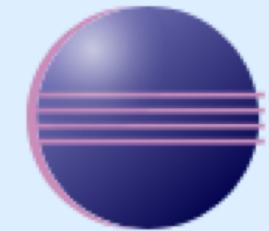
Top Android Development IDE



Android Studio



Visual Studio



Eclipse



IntelliJ IDEA



Cordova

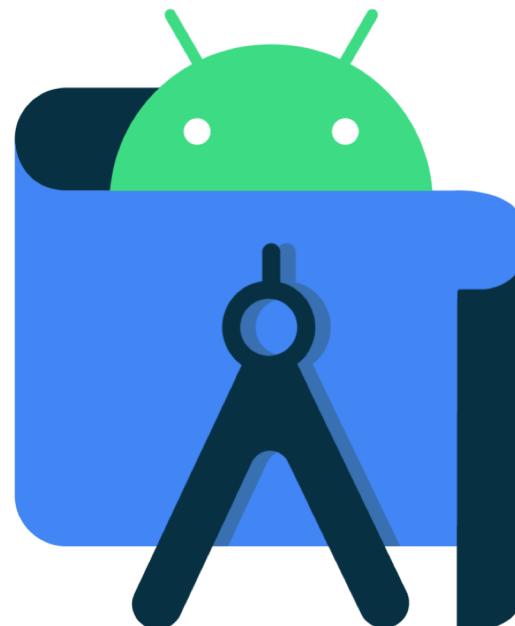


NetBeans



AIDE

- ❑ Android Studio là IDE (môi trường phát triển tích hợp - Integrated Development Environment).
 - ❖ Miễn phí
 - ❖ Phiên bản: Windows, Mac OS X và Linux.
 - ❖ Tích hợp Android SDK hỗ trợ cho việc lập trình trên nền tảng Android như trình biên dịch, các thư viện, máy ảo, tài liệu, source code mẫu



- ❑ Yêu cầu hệ thống
 - ❖ Ram tối thiểu 8GB.
 - ❖ Dung lượng ổ đĩa trống tối thiểu 8GB (IDE + Android SDK + Android Emulator)
 - ❖ Kích thước màn hình tối thiểu 1280 x 800.
- ❑ Tải và cài đặt Android Studio
 - ❖ <https://developer.android.com/studio/install.html>
- ❑ Phiên bản
 - ❖ Electric Eel | 2022.1.1
 - ❖ Dolphin | 2021.3.1
 - ❖ Chipmuk | 2021.2.1
 - ❖ ...



Dolphin | 2021.3.1



THỰC HIỆN CÀI ĐẶT ANDROID STUDIO

Thực hiện theo hướng dẫn của bài 01 – lab 01



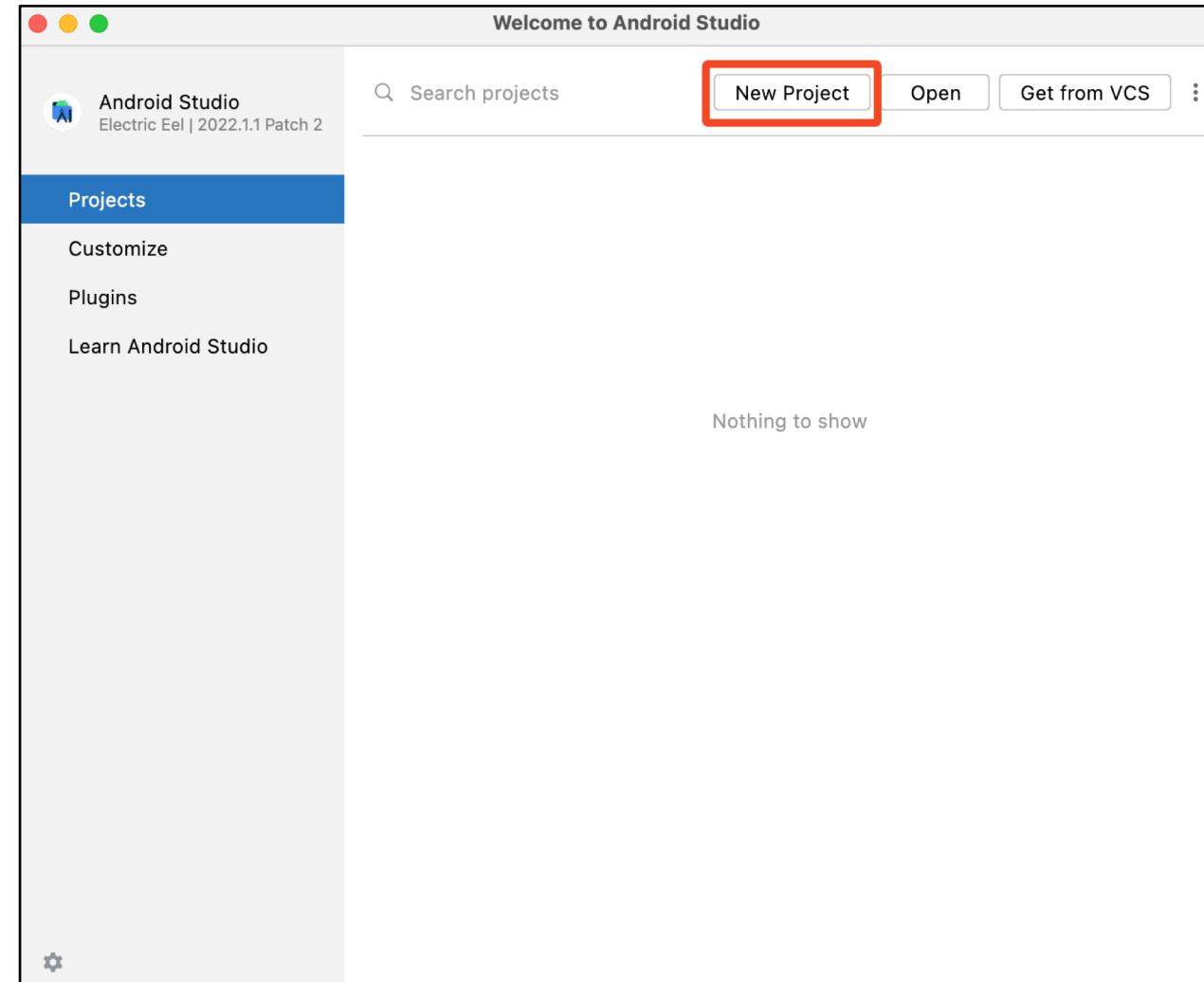
LẬP TRÌNH ANDROID 1

BÀI 1.2: TỔNG QUAN VỀ ANDROID (TT)



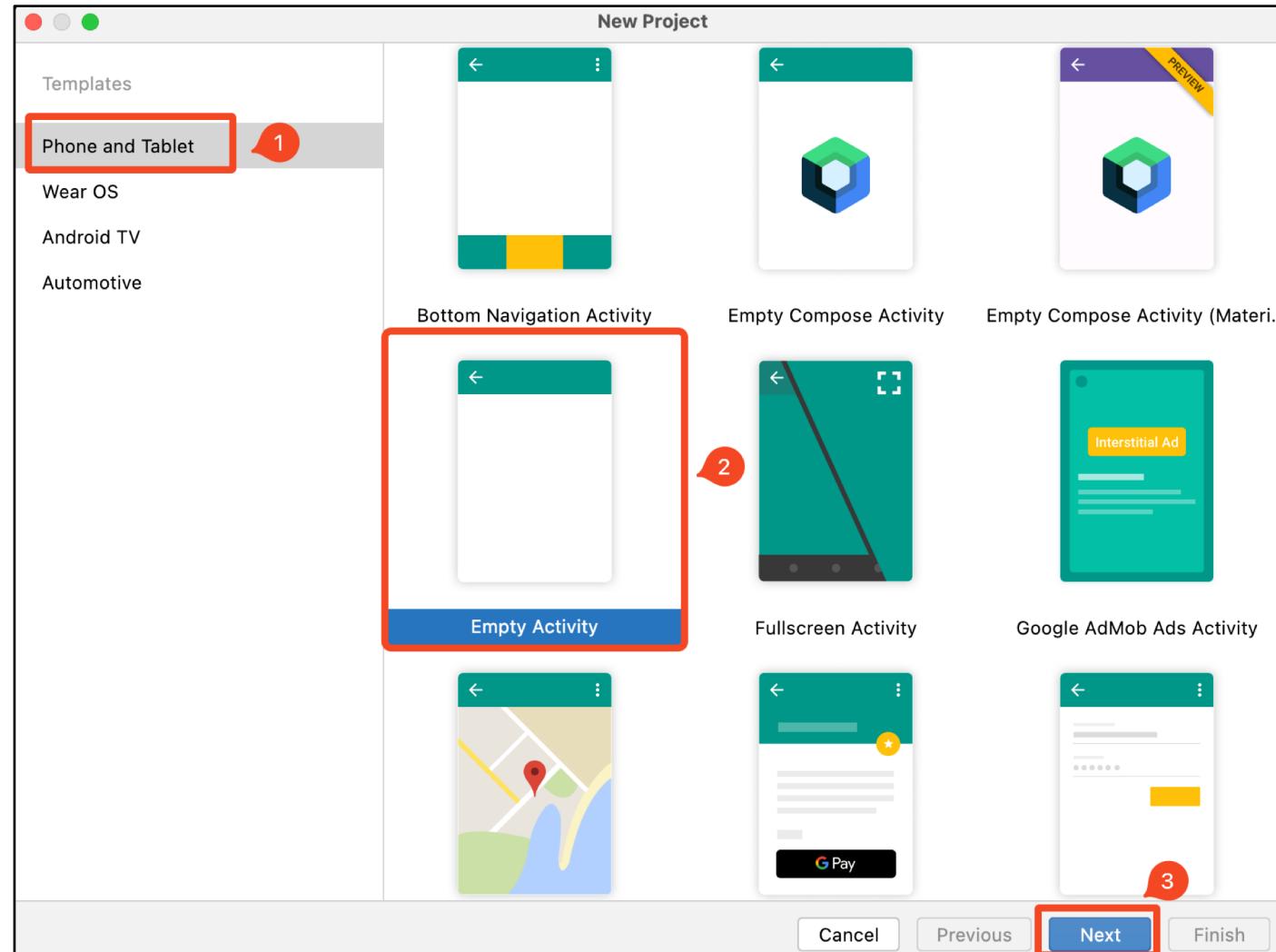
TẠO PROJECT TRONG ANDROID STUDIO

☐ Bước 1: Mở phần mềm **Android Studio** → chọn **New Project**



TAO PROJECT TRONG ANDROID STUDIO

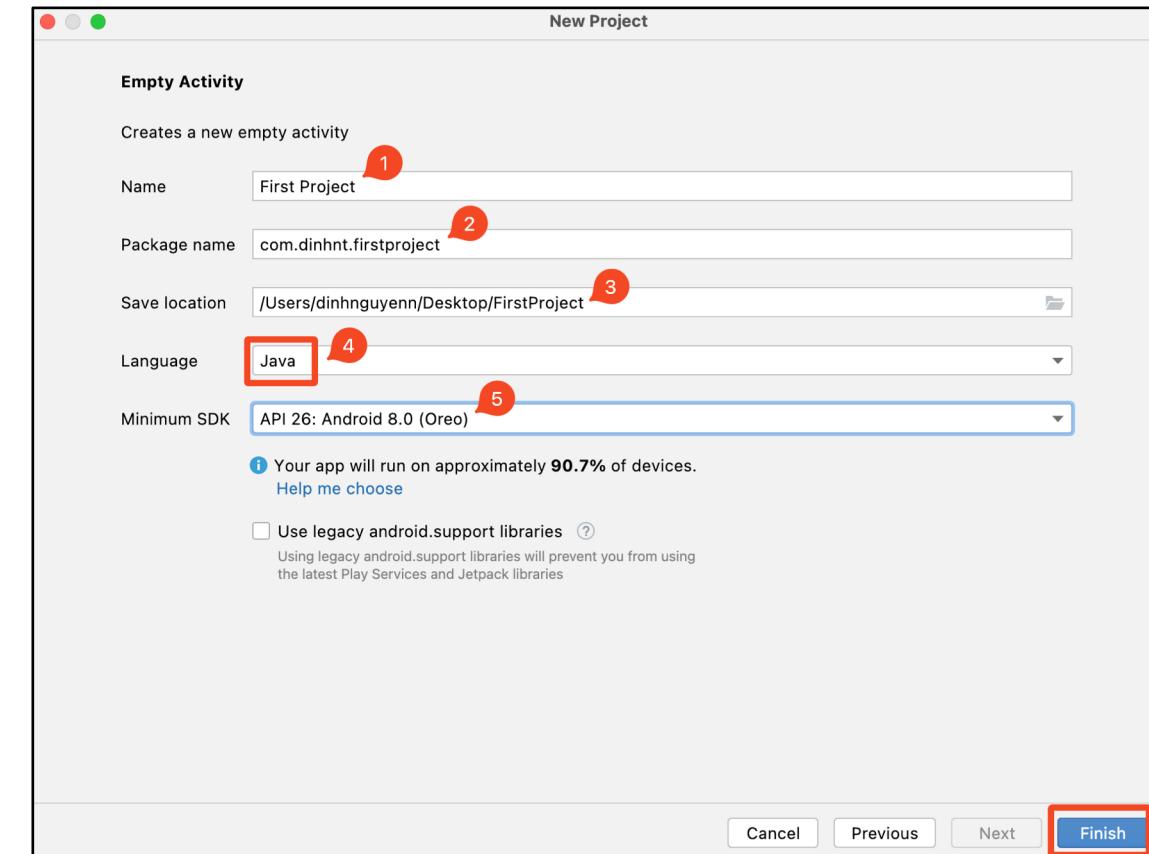
☐ Bước 2: Chọn templates **Phone and Tablet** → chọn **Empty Activity** → chọn **Next**



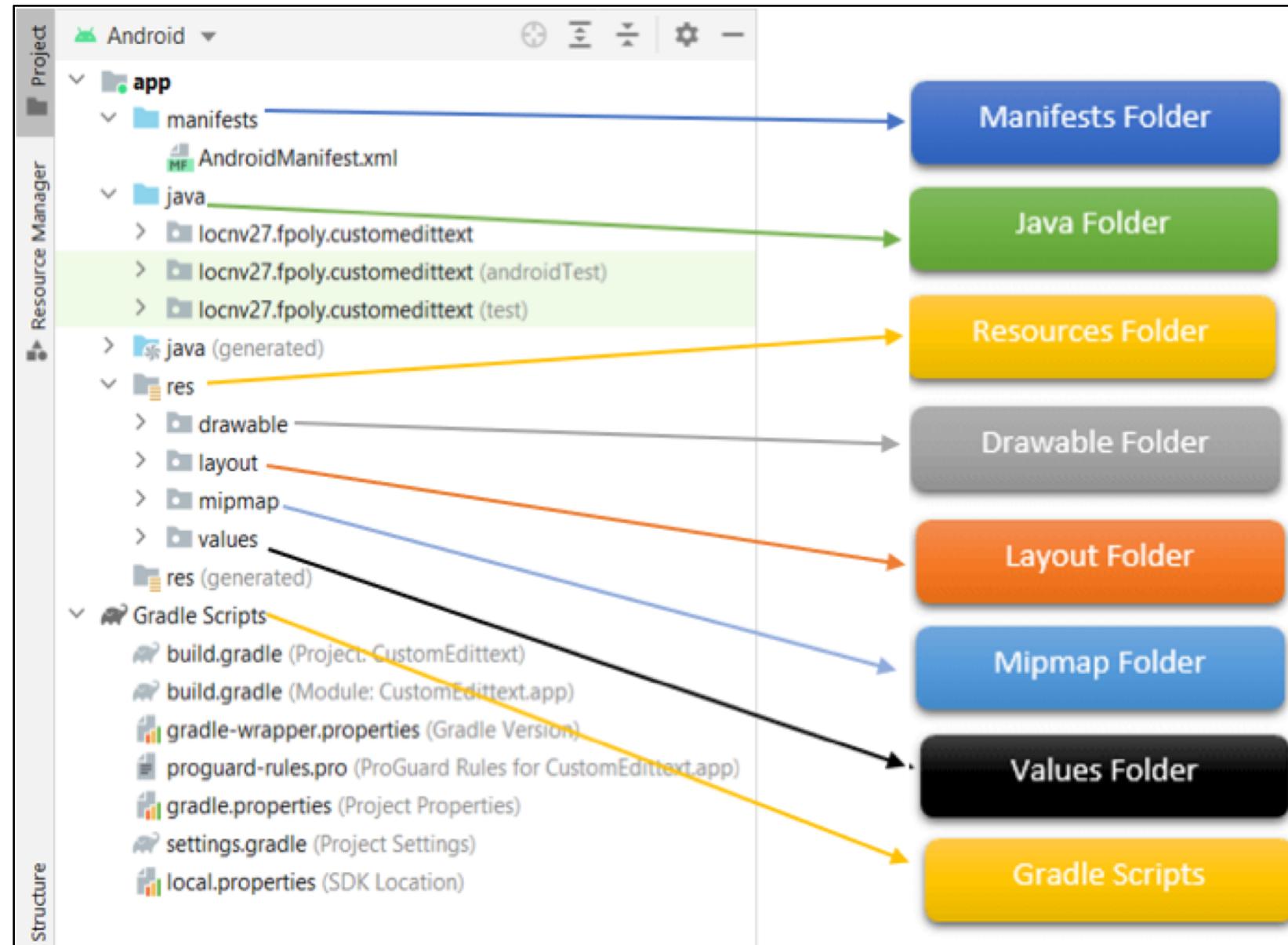
❑ Bước 2: Cấu hình cho project

- ❖ **Name:** Tên project
- ❖ **Package name:** tên package
- ❖ **Save location:** nơi lưu trữ project
- ❖ **Language:** ngôn ngữ code chính trong project, ở đây chúng ta chọn **Java**
- ❖ **Minimum SDK:** phiên bản Android tối thiểu để chạy được ứng dụng

❑ Sau đó nhấn **Finish**

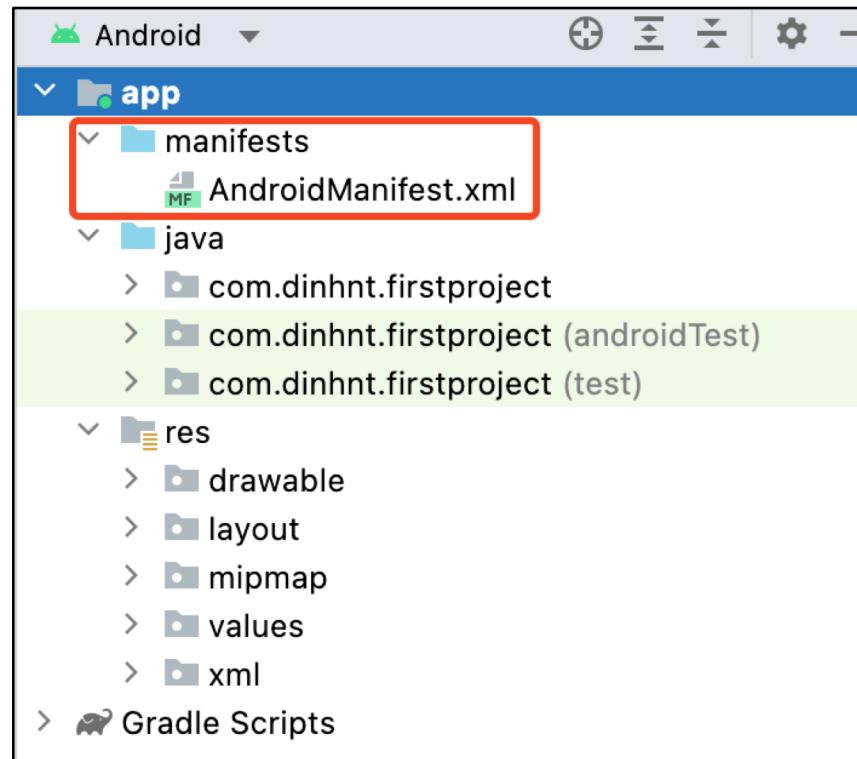


CẤU TRÚC PROJECT TRONG ANDROID STUDIO



❑ Thư mục Manifests

- ❖ Thư mục này sẽ chứa một file Manifest (**AndroidManifest.xml**) cho ứng dụng Android. File manifest này sẽ chứa thông tin về ứng dụng, chẳng hạn như android version, access permissions, metadata, v.v và các component của ứng dụng. File manifest sẽ hoạt động như một trung gian giữa hệ điều hành Android và ứng dụng.



☐ Thư mục Java

- ❖ Thư mục này sẽ chứa tất cả các file mã nguồn java (*.java) mà chúng ta sẽ tạo trong quá trình phát triển ứng dụng, bao gồm cả mã **JUnit test code**. Bất cứ khi nào tạo bất kỳ project / ứng dụng mới nào, file lớp **MainActivity.java** sẽ tự động tạo trong package là *com.dinhnt.firstproject*, như hình bên dưới.

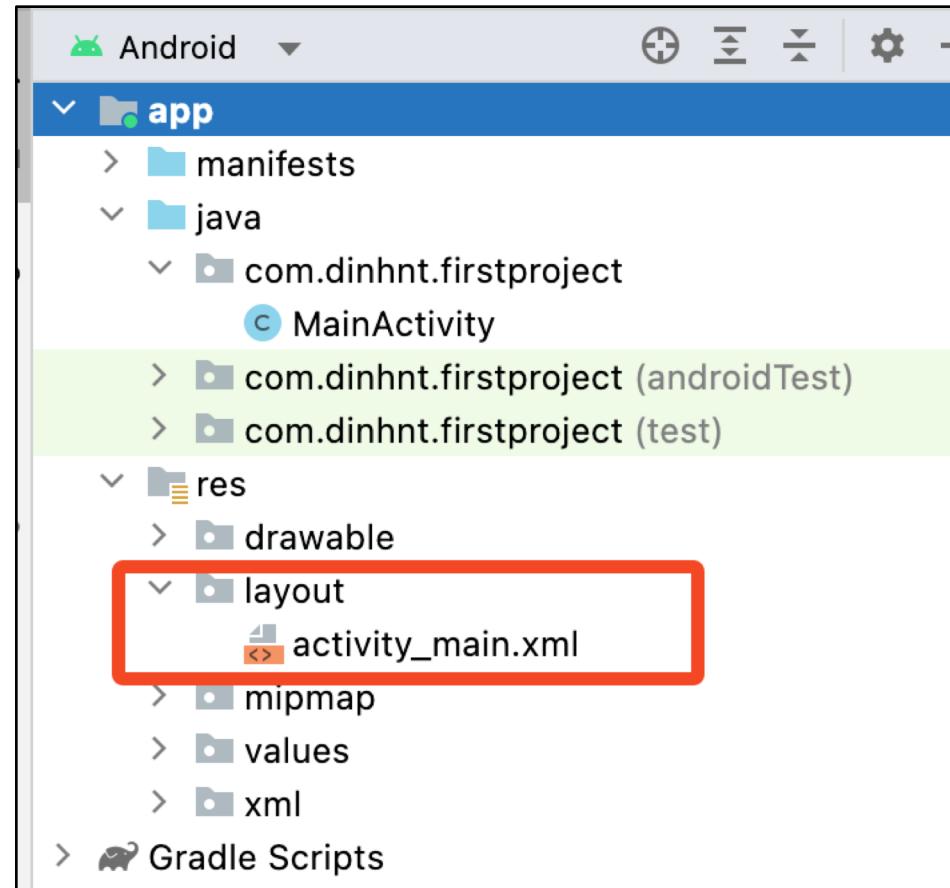
The screenshot shows the Android Studio interface. On the left, the Project Navigational Bar displays the project structure:

- app (selected)
- manifests
- java
 - com.dinhnt.firstproject
 - MainActivity (selected)
 - com.dinhnt.firstproject (androidTest)
 - com.dinhnt.firstproject (test)
- res
 - drawable
 - layout
 - mipmap
 - values
 - xml
- Gradle Scripts

A red box highlights the `MainActivity` file in the `java/com.dinhnt.firstproject` folder. On the right, the code editor shows the `MainActivity.java` file content:package com.dinhnt.firstproject;
import ...
2 usages
public class MainActivity extends AppCompatActivity {
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);
 }
}

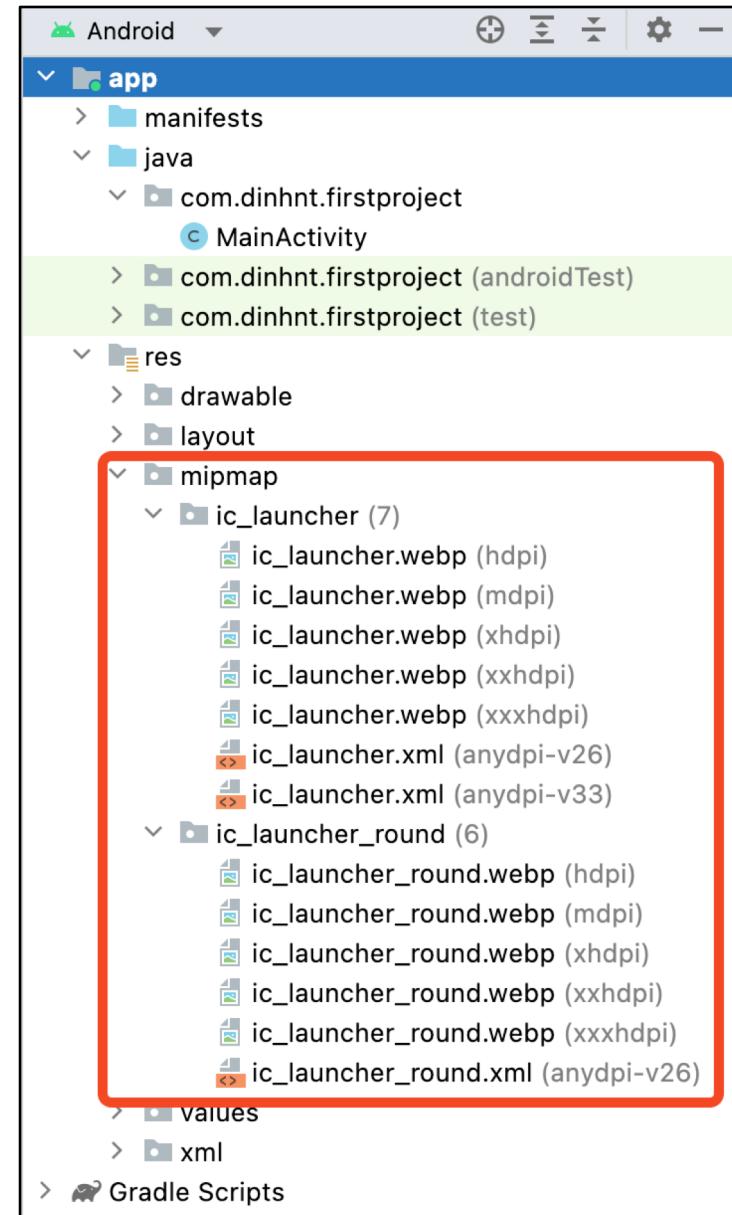
❑ Thư mục layout (res/layout)

- ❖ Thư mục này sẽ chứa tất cả các file XML layout đã sử dụng để xác định Giao diện người dùng của ứng dụng. Sau đây là cấu trúc của thư mục layout.



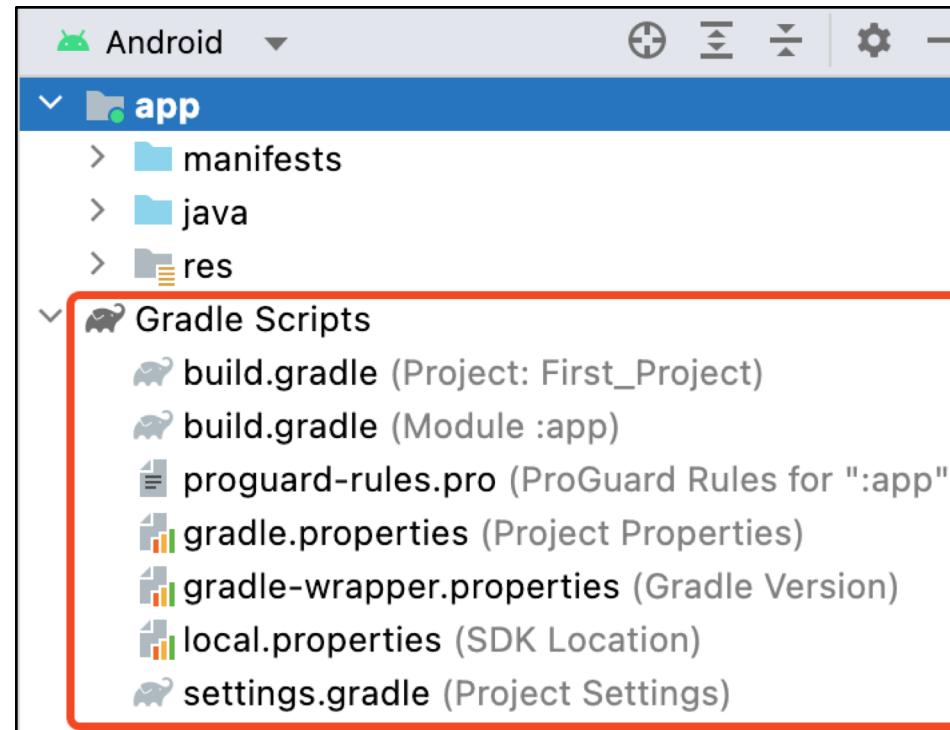
❑ Thư mục Mipmap (res / mipmap)

- ❖ Thư mục này sẽ chứa các biểu tượng ứng dụng/launcher được sử dụng để hiển thị trên màn hình chính. Các loại biểu tượng sẽ có tỷ trọng khác nhau như **hdpi**, **mdpi**, **xhdpi**, **xxhdpi**, **xxxhdpi**, để sử dụng dựa trên kích thước của thiết bị.



❑ Gradle Scripts

- ❖ Trong Android, Gradle là công cụ build hệ thống và Gradle được tích hợp sẵn vào Android Studio, và được điều khiển một cách tự động thông qua Android Studio.
- ❖ Trong gradle có **build.gradle (Project)** và **build.gradle (Module)** được sử dụng để build các cấu hình áp dụng cho tất cả các module ứng dụng hoặc dành riêng cho một mô-đun ứng dụng.





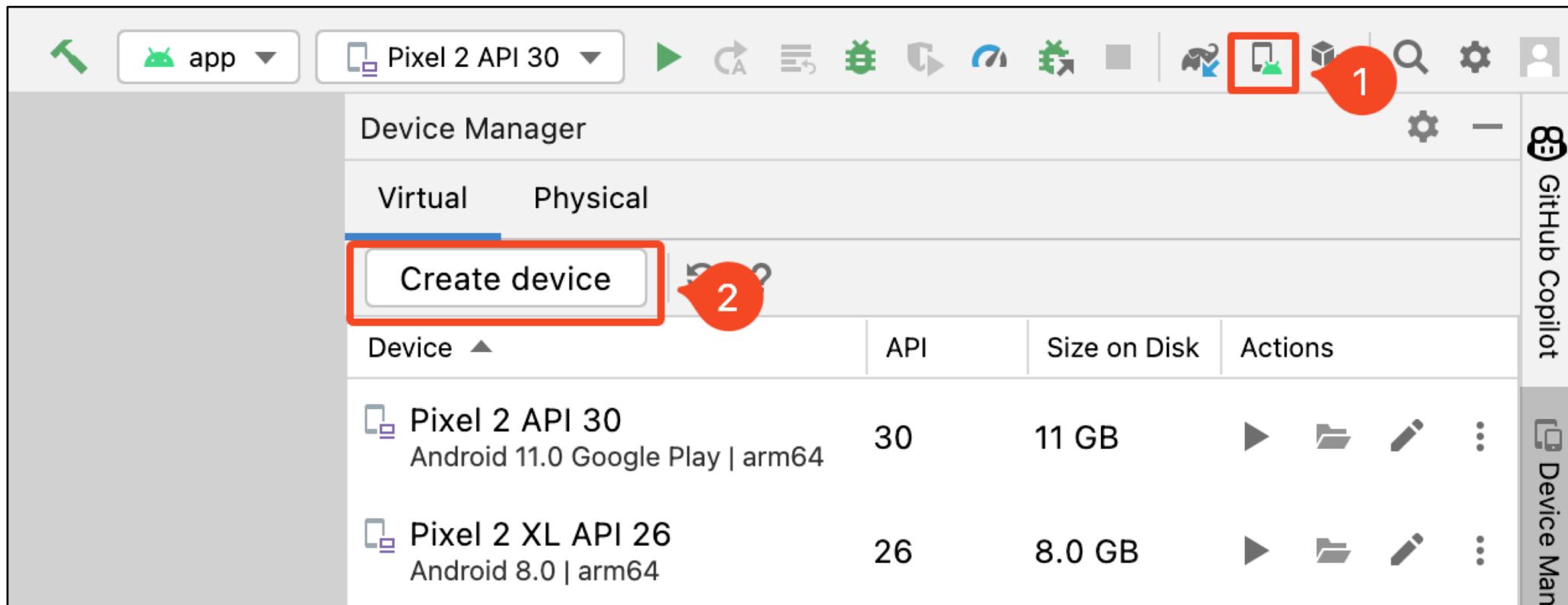
TẠO MÁY ẢO TRONG ANDROID STUDIO

TẠO MÁY ẢO TRONG ANDROID STUDIO

Chọn biểu tượng

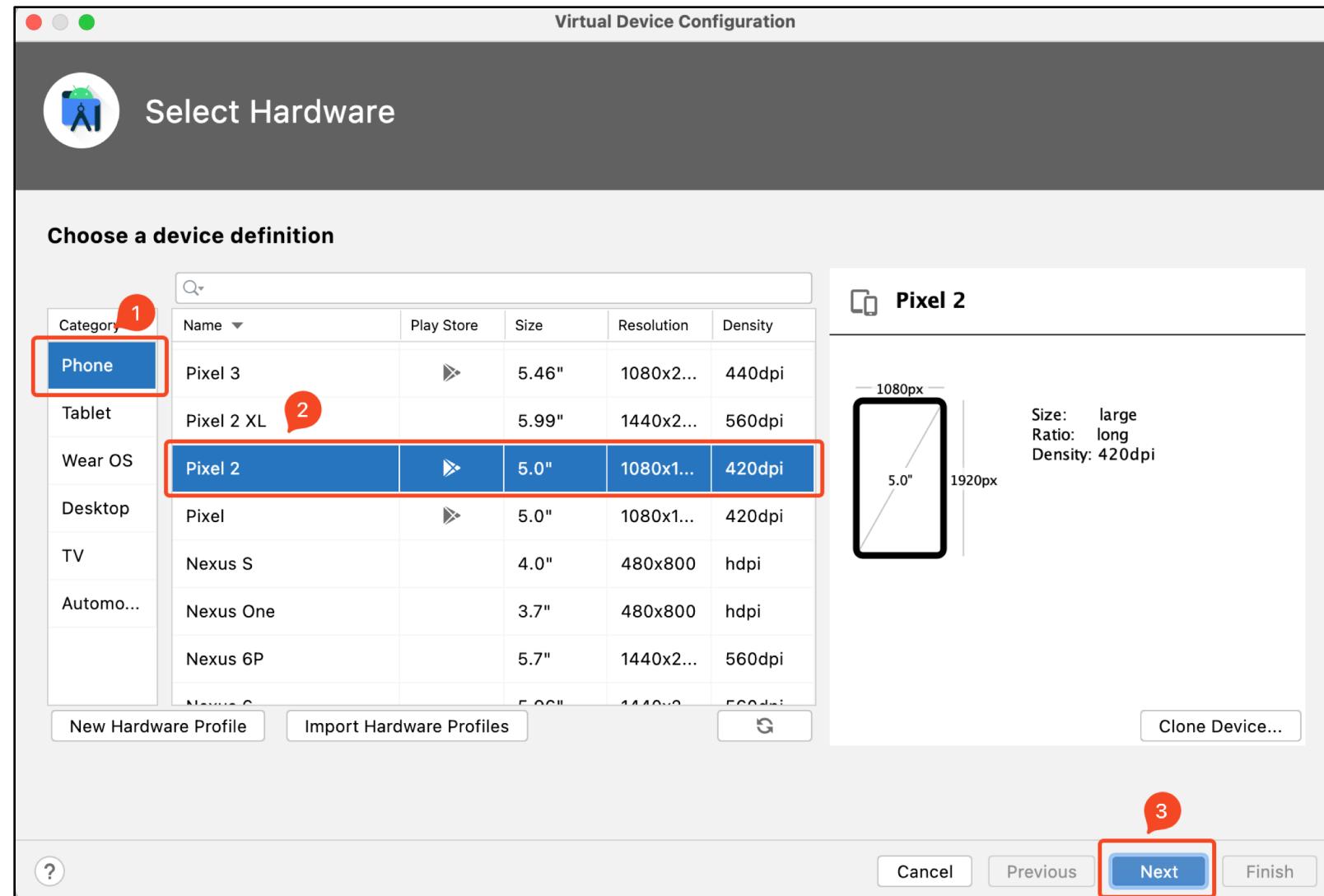


và chọn **Create device**

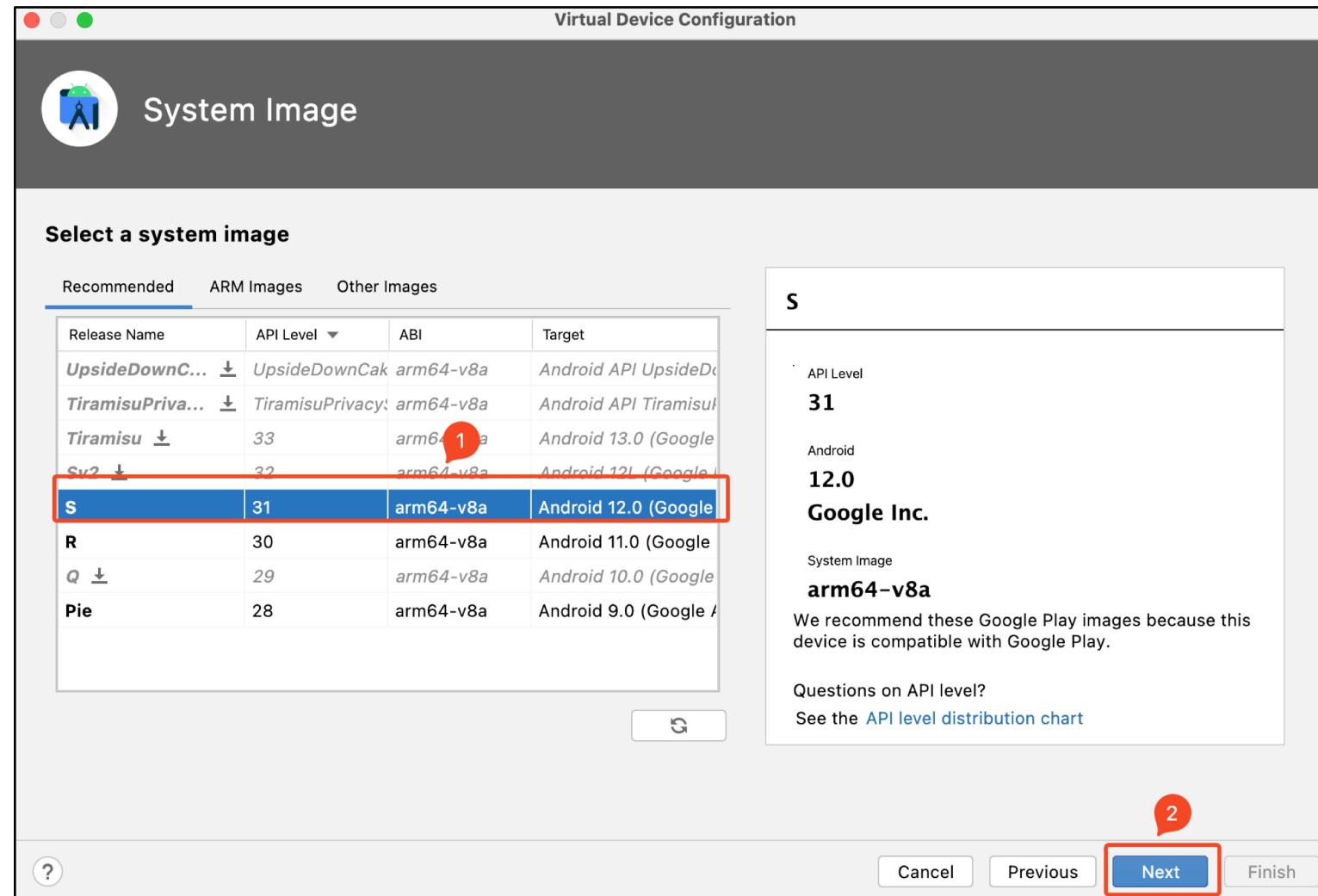


TẠO MÁY ẢO TRONG ANDROID STUDIO

Chọn loại máy ảo phù hợp

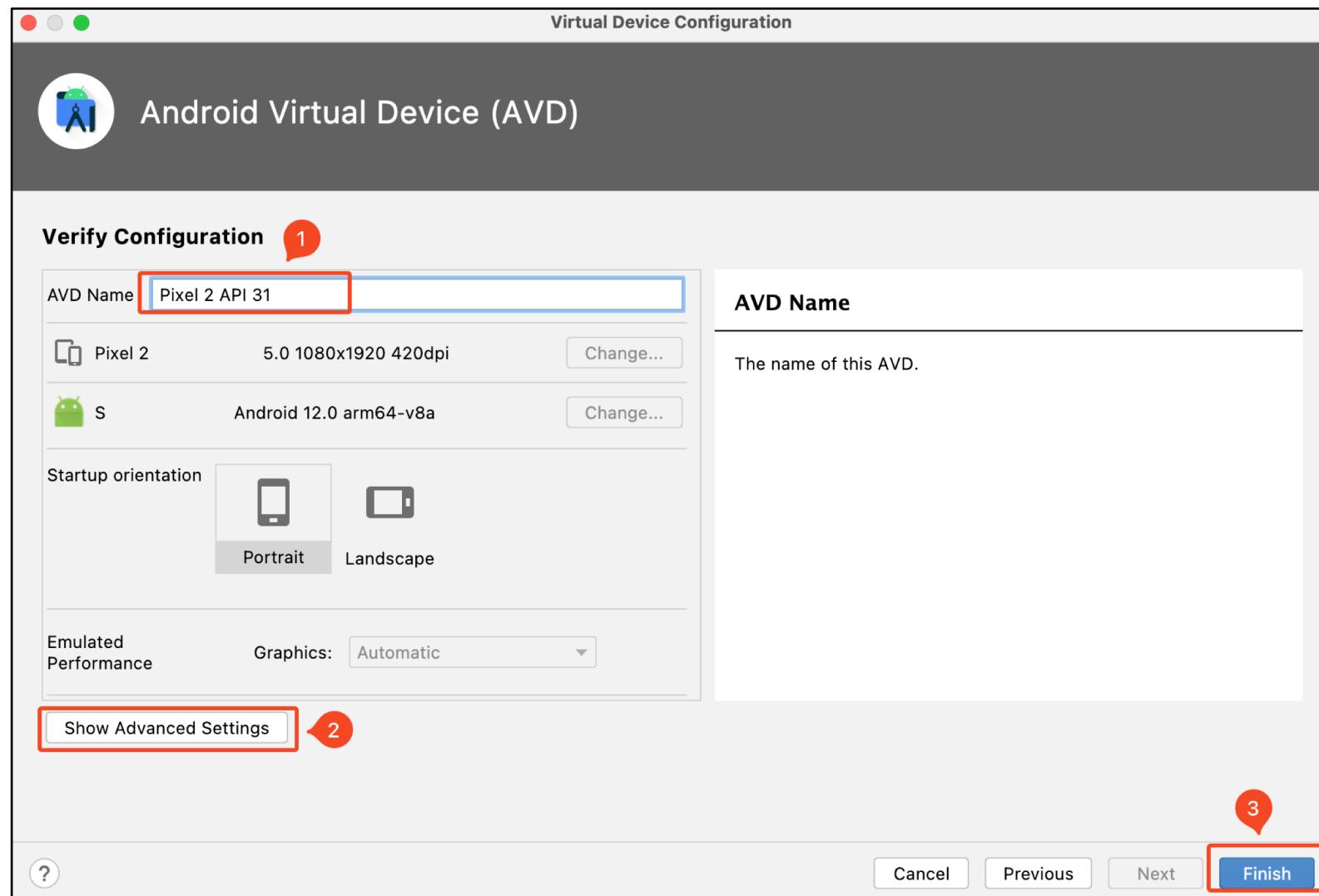


Chọn phiên bản Android phù hợp với máy ảo

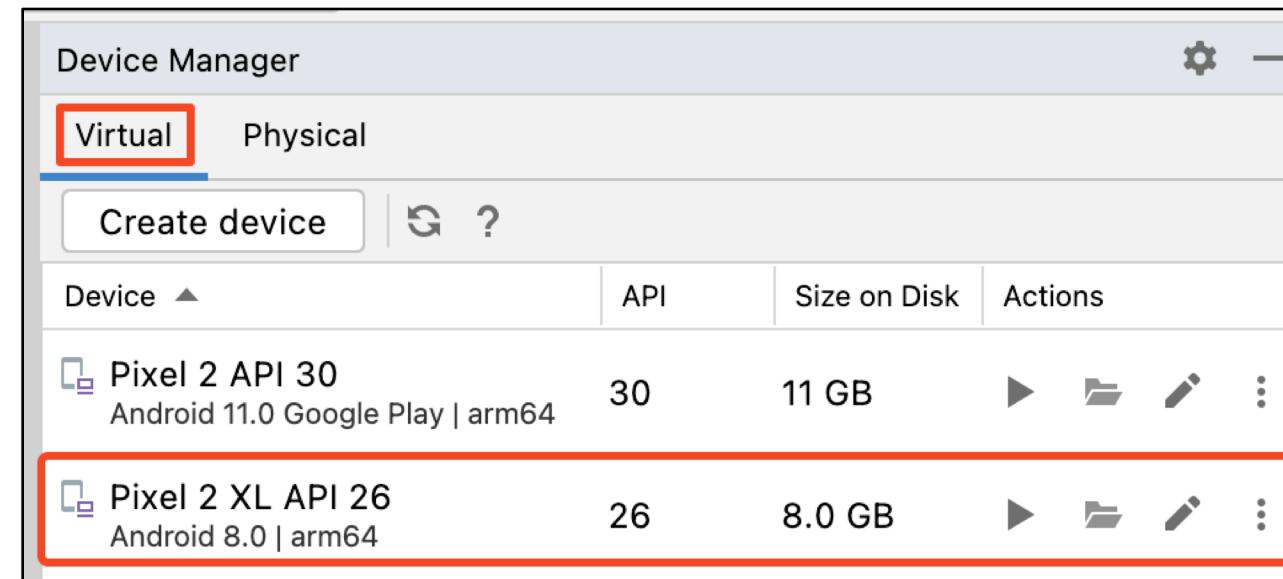


TẠO MÁY ẢO TRONG ANDROID STUDIO

- Đặt tên cho máy ảo, có thể cấu hình thêm cho máy ảo (nếu muốn). Sau đó nhấn **Finish**



- Sau khi cài đặt thành công, máy ảo sẽ được hiển thị trong **Device Manager (Virtual)**





THỰC HIỆN BÀI LAB SỐ 2 VÀ BÀI LAB SỐ 3



FPT
Education

FPT POLYTECHNIC

Thank you